

## 09 OOP - 練習問題

作成者 : GT F

最終更新日 : 2019-08-15

### 質問 :

質問 1 : クラス (Class) とは ? クラス定義用のキーワードは「`class`」である。

質問 2 : クラスメンバ (Class member) の何種類ありますか。

1. 属性・フィールド (Field)
2. 関数 (Function) ・ メソッド (Method) ・ 操作 (Operator)

質問 3 : クラスメンバの関数は特殊の関数「コンストラクタ」が持っています。

質問 4 : コンストラクタの特徴

1. 戻り値不要
2. 関数名 = クラス名称

質問 5 : コンストラクタの作用 : クラス属性の初期化する。

質問 6 : 1 つクラスに複数のコンストラクタ持つ可能。

質問 7 : キーワード `this` の使用方法 : `this.name` ? `name` ?

```
1 public class People {  
2     private String name;  
3     public void setName (String name) {  
4         this.name = name;  
5     }  
6 }
```

質問 8 : クラスを利用する為、インスタンス化する必要があります。インスタンスの演算子は `new` です。

質問 9 : 演算子 `new` の直後呼び出しているメソッドは「コンストラクタ」 ;

```
1 People obj = new People();
```

質問 10 : アクセス制御の 3 種類あります。

1. + public
2. # protected
3. - private

質問 11 : 静的のメンバの定義するキーワード (修飾子) は `static`。

質問 1 2 : 静的のメンバアクセス時、インスタンス化不要。アクセス方法は「クラス名. 静的のメンバ」。

```
1 System.out.println("1234"); // 静的のField
2 String value = String.format("%05d", 0); // 静的のメソッド
```

質問 1 3 : クラス「Animal」を定義して、インスタンス化しオブジェクト cat を作成します。変数catのsayHello()メソッドを呼び出してください。

1. 属性 : 年齢
2. 属性 : 名前
3. 属性 : 分類
4. コンストラクタ : 名前を渡して、初期化する
5. 操作 : sayHello() : void
6. 操作 : getName(): String 自分の名称を返す

質問 1 4 : 以下 1 行を解釈してください。

```
1 System.out.println(new Animal("cat").getName());
```

質問 1 5 : クラス継承する用キーワードは「`extends`」？

質問 1 6 : 継承しているクラス。上位クラスは「親クラス」・「スーパークラス」。下位クラスは「サブクラス」と呼びます。

質問 1 7 : サブクラス&アクセス制御（`public`, `protected`）。

質問 1 8 : サブクラスと親クラス同じメソッド名かつメソッドの引数は一致する場合、「オーバーライド」となります。

質問 1 9 : 以下Animalクラスを継承して以下 2 クラスを作成してください。

1. クラス Cat。sayHelloメソッドをオーバーライドして、System.out.println("ニャー");
2. クラス Dog。sayHelloメソッドをオーバーライドして、System.out.println("ワン");

※親クラスに明示的なコンストラクタが存在しない場合、サブクラスは必ず親クラスのコンストラクタをオーバーライドします。

質問 2 0 : 以下コードをIS-A関係で説明してください。

```
1 Animal v0 = new Animal(); // v0 は Animal 型
2 Animal v1 = new Cat("a cat"); // v1 は Cat 型, v1 は Animal 型
3 Animal v2 = new Dog("a dog"); // v2 は Dog 型, v2 は Animal 型
4
5 Cat v3 = new Cat(""); // OK
6 Dog v4 = new Dog(""); // OK
7
8 Cat v5 = new Animal(""); // NG
```

# 覚えておこう

## メソッド呼び出す

1. メソッドの所属クラスを確認する。例：JFrameクラスのsetVisibleメソッド。
2. メソッドは静的、静的ではないかを確認。
3. 静的の場合：直後クラス名 + `.` で呼び出す。
4. 非静的の場合：クラスをインスタンス化する `new`。
5. インスタンス化するのはコンストラクタを呼び出す。コンストラクタの必要な引数を用意する。
6. インスタンス化した後取得したオブジェクト + `.` でメソッドを呼び出す。

## 用語

 いいね 1 番に「いいね」しましょう

ラベルがありません 